

AWX CLI

1- Setup the CLI

We need to install the AWX CLI is through pip directly from PyPI:

```
pip3 install awxkit
awx --help
```

In order to authenticate to the awx cli, use the following commands:

```
export TOWER_HOST=http://51.254.116.97:8052/
$(TOWER_USERNAME=admin TOWER_PASSWORD=password awx login -f human)
awx config
```

Next, let's create a new organization for the lab, name it "vbs":

```
$ awx organization create --name "vbs" --max_hosts 100
```

Remember how you used the awx help to get down to the needed command.

2- Create an Inventory

The first step is to create a static inventory; dynamic inventories will be covered later. Using the awx command, create an inventory named "Example Inventory". You can work with multiple organizations in Tower. In this lab we'll work in the **VBS** organization.

```
$ awx -f human inventory create --name "Example" --organization "vbs"
```

3- Add Hosts to the Inventory using awx

Now that we have the empty inventory created, add your two managed hosts using their internal hostnames node1 and node2, again using awx.

```
$ awx -f human host create --name "node1" --inventory "Example"
$ awx -f human host create --name "node2" --inventory "Example"
```

4- Create Machine Credentials

SSH keys have already been created and distributed in your lab environment having password-less login. We want to configure these credentials to access our managed hosts from AWX.

First we need to look for the Id of the credentials type “Machine” using this request:

```
$ awx -f human credential_types list
```

```
id name
== =====
5  Amazon Web Services
15 Ansible Tower
9  Google Compute Engine
10 Microsoft Azure Resource Manager
6  OpenStack
8  Red Hat Satellite 6
14 Red Hat Virtualization
7  VMware vCenter
18 CyberArk AIM Central Credential Provider Lookup
20 CyberArk Conjur Secret Lookup
21 HashiCorp Vault Secret Lookup
22 HashiCorp Vault Signed SSH
19 Microsoft Azure Key Vault
17 Ansible Galaxy/Automation Hub API Token
13 Insights
16 OpenShift or Kubernetes API Bearer Token
4  Network
2  Source Control
1  Machine
11 GitHub Personal Access Token
12 GitLab Personal Access Token
3  Vault
```

Then issue the following command to create the new machine credentials:

```
$ awx -f human credential create --name "Example Credentials" \
  --organization "vbs" \
  --credential_type 1 \
  --inputs '{"username": "labadm", "ssh_key_data": "@~/.ssh/id_rsa}"'
```

You have to ensure that you are passing a valid ssh private key in your inputs, otherwise you will face a key validation error like this:

```
{"inputs": {"ssh_key_data": ["Invalid certificate or key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCS3ErS9Tu/7L6vjFSZF/U4g9nq2c1Nx+I19cJclNmE/gFoo9WxGbXWlsW0qRN4.\n."}]}}
```

5- Create the Project

The Ansible content used in this lab is hosted on Github. The next step is to add a project to import the playbooks:

```
$ awx -f human project create --name="Apache" \
  --scm_type=git \
  --scm_url="https://github.com/ghazelatech/ansible-labs-playbooks.git" \
  --organization "vbs" \
  --scm_clean=true \
  --scm_delete_on_update=true \
  --scm_update_on_launch=true \
  --wait
```

6- Create a Job Template

Before running an Ansible Job from your AWX instance you must create a Job Template. Here AWX CLI will work on the resource **job_template**:

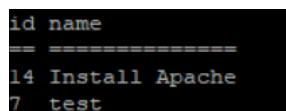
```
$ awx -f human job_templates create \
  --name="Install Apache" \
  --inventory="Example" \
  --project=Apache \
  --playbook=apache_install.yml \
  --become_enabled="yes"
```

7- Attach Credentials to Job Template

Finally we need to attach the credentials we created above to the Job Template.

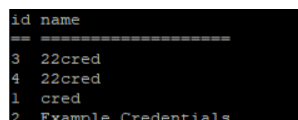
We need first to retrieve the credential Id using this command:

```
$ awx -f human credentials list --query "{id: id, name: name}"
```



id	name
14	Install Apache
7	test

```
$ awx -f human job_templates list --query "{id: id, name: name}"
```



id	name
3	22cred
4	22cred
1	cred
2	Example Credentials

Then we could associate the credential to the job template like below:

```
$ awx job_template associate --credential 2 --job-template 14
```